UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/750,188	12/29/2000	Thomas S. Brady	3175-51	5835

21013 7590 11/05/2003

AGFA CORPORATION
LAW & PATENT DEPARTMENT
200 BALLARDVALE STREET
WILMINGTON, MA 01887

EXAMINER	
SIANGCHIN, KEVIN	
ART UNIT	PAPER NUMBER
2623	6

DATE MAILED: 11/05/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/750,188	**Applicant(s)** BRADY, THOMAS S.	
	Examiner Kevin Siangchin	**Art Unit** 2623	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 December 2000* .

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 December 2000* is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)□ The proposed drawing correction filed on _____ is: a)□ approved b)□ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)□ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____ .

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) □ The translation of the foreign language provisional application has been received.

15)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) □ Interview Summary (PTO-413) Paper No(s). _____ .
5) □ Notice of Informal Patent Application (PTO-152)
6) □ Other: .

Drawings

1. New corrected drawings are required in this application because the drawings are not of sufficient quality to permit publication. Applicant is advised to employ the services of a competent patent draftsperson outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance.

2. The drawings are objected to under 37 CFR 1.83(a) because they fail to show:

i. the logical steps of the disclosed AGFA compression technique that relate to the applicant's claimed invention;

ii. the various data structures that the applicant mentions in the specification; in particular, the 9, 10, 11, or 12 bit blocks, that contain the continuation bit or code, used to represent the repeat-count;

iii. the way in which these "repeat count multiple output codes" (see page 18, line 20 of the applicant's disclosure) are "strung together" to represent the repeat-count;

iv. how the applicant proposes that the claimed compression method be incorporated into the framework AGFA compression technique described in the applicant's specification;

and so on, as described in the specification. Any structural detail that is essential for a proper understanding of the disclosed invention should be shown in the drawing. MPEP § 608.02(d). A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The inclusion, for example, of flow diagram(s) and/or state diagram(s), as well as more illustrative versions of the current block diagrams, may clarify some of these details. The objection to the drawings will not be held in abeyance.

3. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. The following are features of the claims which are missing from the applicant's attached drawings.

i. The claims propose a method(s) of compressing image information (claims 7-13), or an encoder (claims 1-6, 19-20) or imaging system (claims 14-18) that implement that method. This method of compressing image information is, indeed, the essential aspect of the applicant's claimed invention. However, the drawings attached fail to illustrate, in any way, that aspect of the applicant's claims.

ii. The various states of data in the method, encoder, and imaging system described in the applicant's claims (e.g. uncompressed/un-encoded image data, compressed/encoded image data, decompressed/decoded image data) are not shown.

iii. The various data structures used to represent the sequence of characters representing an image (e.g. continuation codes, predefined compression codes, etc.)

Therefore, the preceding aspects of the claims must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

4. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Specification

5. The disclosure is objected to because of the following informalities:

i. The applicant makes reference to a first, second, third and fourth sequence of characters throughout the disclosure. For example, from page 7, lines 15-35, and page 8, line 1 the applicant recites:

> In a preferred implementation, the processor receives a <u>first sequence of characters</u> representing an image. A first character in the sequence is read. The processor determines if the read character represents the white or black image data. If so, one or more characters occurring immediately subsequent to the first character in the sequence of characters are read. The processor determines if the one or more characters match the read first character and, if so, generates a <u>second sequence of characters</u>, including the stored predefined compression code, representing the matching one or more characters.
>
> Preferably, the memory stores two predefined compression codes corresponding respectively to white image data and black image data. In such an implementation, the processor may receive a <u>third sequence of characters</u> representing the image, read a first

> character in the third sequence, and determine if the read character in the third sequence represents white or black image data. If so, the processor reads one or more characters occurring immediately subsequent to the first character in the third sequence of characters, and determines if these read characters match the first character in the third sequence. If so, the processor generates a fourth sequence of characters, including the applicable stored predefined compression code, representing the matching characters in the third sequence.

Note that the phrases "first sequence of characters", "second sequence of characters", "third sequence of characters" and "fourth sequence of characters" have been emphasized. There are instances in the applicant's disclosure when the applicant is inconsistent with his usage of these "first", "second", "third" and "fourth" distinctions. For example, observe that on page 8, lines 20-32, the applicant states, "…if the number of repeating characters is very large … a second code will be required to complete the encoding. Therefore, the processor generates a third sequence of characters …". That third sequence is clearly different from the third sequence mentioned in the above excerpt from page 7, lines 15-35, since the former is being generated by the processor, whereas the latter is being received by the processor. The applicant may benefit from the usage of a more unambiguous notation or method to distinguish the different sequences of characters described in the disclosure. Note that the applicant's usage of the phrases "first sequence of characters", "second sequence of characters", "third sequence of characters" and "fourth sequence of characters" is not limited to the sections of the disclosure mentioned above. The disclosure is replete with instances where sequences of characters are referred to in this manner.

ii. The following sentence is from page 13, lines 1-5:

> Although, the image discussed with reference to the page assembly mode may be the same as the image discussed with reference to the prior page assembly mode, conversion in the page assembly mode will typically result in even a greater amount of encoded data than the conversion in the previously discussed banding mode.

The phrase "prior page assembly mode" underlined above appears to a typographical error. Given the context in which this phrase appears and the ambiguity as to what prior page assembly mode to which the applicant is referring, it is suggested that the applicant replace "prior page assembly mode" with "banding mode". This would be consistent with previous and subsequent portions of the disclosure.

iii. Throughout the disclosure, reference is made to the "pack-bit" compression technique. As described in the specification, this technique is essentially the same, in principle, to the well-known "PackBits" method of compression. Indeed, the pack-bit method described by the applicant and the aforementioned PackBits method differs only in name. It would, therefore, be preferable, in the interest of clarity, that the applicant uses the more recognizable and more commonly used name, "PackBits", when referring to this compression technique. Note that name PackBits (or PB, where appropriate) will be used, henceforth in this document, to designate the applicant's pack-bit method.

Appropriate correction is required.

Claims

6. Claim 1-20 are objected to because of the following informalities:

i. Claims 1 and 7 contain the following passage (note that this passage was taken from claim 1 and the wording of claim 7 is nearly identical): "...to determine that the read first character corresponds to the one of the white and the black image data, to read one or more characters immediately subsequent to the first character in the ... sequence of characters, to determine that the read one or more characters match the read first character ...". (Incidentally, the phrase "to the one of the white and the black image data" should be revised to correct the obvious grammatical errors). This passage, phrased as such, can be interpreted as not corresponding to the same process, which is described in the following excerpts from page 14, lines 9-17:

> ... as a stream of sequential data is processed prior to encoding if, at the start of the sequence, the immediately preceding character, which is yet to be encoded, matches the next character in the stream and this next character is either solid black or solid white,

> and hence digitally represented in binary form by all 1's or 0's, encoding is interrupted. During the interruption, a determination is made as to whether the one or more characters, immediately following the next character in the sequence, also match the next character.

; and from page 17, lines 8-15:

> Also, because the "0" is recognized as special, the RIP processor 2050a, automatically scans ahead to read the next character in the sequence to determine if it matches with the initial "0" in the sequence. If not, the scanning ahead is immediately discontinued and the RIP processor 2050a proceeds with normal processing. If so, the scanning ahead continues on a character by character basis until no match with "0" is found, at which point the scanning ahead is discontinued and normal processing continues.

These excerpts relate to the method of claim 7 and the encoder of claim 1. In these excerpts, a first character is read. If that first character represents either black or white image data, it is then determined whether there is a sequence of matching characters immediately subsequent to that first character. Furthermore, as described in the second of the preceding excerpts, this process ensures that the longest sequence of matching characters that immediately follow the said first character is determined, if such a sequence exists at all. The process or method, described in the above passage from claims 1 and 7, can be interpreted differently as follows. According to that passage, the determination made with regard to the one or more characters immediately following the first character is whether or not the read one or more characters immediately following the read first character match the said first character. One of ordinary skill in the art may conclude that the result of this determination may differ, in certain instances, from that of the analogous determination described in the portions of the specification listed above. Specifically, if the sequence of characters immediately following the first character consists of a sequence of matching characters followed by a sequence of non-matching characters, then the result of the determination of claims 1 and 7 will differ from the result of the determination described in the specification. For example, if the first read character is ‘0’ and the sequence of characters immediately following is, ‘0’, ‘0’, ‘0’, ‘0’,’2’,’3’,’4’, then clearly the sequence of one or more characters immediately subsequent to the read first character does not match the first character, even though there is clearly a sequence of matching characters immediately following the read first character. In this regard, the said determination, according to the claims 1 and 7, would indicate that the one or more characters immediately subsequent to the first read character (i.e. ‘0’, ‘0’, ‘0’, ‘0’,’2’,’3’,’4’) fails to match the read first character (i.e. ‘0’), and would, thereby, fail to recognize the sequence of matching characters immediately following the read first character (i.e. ‘0’, ‘0’, ‘0’, ‘0’). The said determination described in the specification, on the other hand, would, in this example, identify the sequence of matching characters immediately following the read first character. By changing the phraseology of the above passage from claims 1 and 7, and the claims where that passage may appear similarly (namely, claims 2, 8, 15, and 20), to reflect its intended meaning as described above and in the applicant’s specification, the applicant would resolve the potential for inconsistent interpretation of the said claims.

Appropriate correction is required.

8. It is assumed that in the appropriate claims the applicant meant that one or more characters are read that occur immediately subsequent to the first read character (where the read one or more characters immediately subsequent to the first character may or may not match the first read character); and the determination made with respect to the read one or more characters immediately subsequent to the first character is whether or not there exists, within these read one or more characters, a sequence of matching characters that occurs immediately subsequent to the first read character.

Rejections under U.S.C. § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-4, 7-11 and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Notenboom (U.S. Patent 4955066).

11. The following is with regard to claim 7. Notenboom discloses a method of compressing a text file in digital form. As shown in Fig. 1 of Notenboom, a full text file is sequentially compressed and assembled into a compressed text file for storage. Notice that during the second pass (Notenboom, Fig. 1, reference number 14), the text is run-length encoded. It should be noted that encoding or compressing data using a run-length encoding method is well known in the art. In Notenboom's method, all runs of between 4 and 255 identical characters are replaced with a run-length flag, the character and a repetition count, in a manner consistent with typical run-length encoding implementations. In one embodiment of Notenboom's method, frequently occurring runs, such as space, are provided with a unique flag (a code identifying the character – see Notenboom, column 1, lines 56-60) followed by a repetition count, thus avoiding the need to provide the character and improving the overall compression ratio. See Notenboom, column 3, lines 33-46. It will be shown below that, in this regard, Notenboom teaches a run-length encoding method, utilizing predefined compression codes in a manner similar to that of the applicant, which can be interpreted as conforming, essentially, to the compression method proposed in the applicant's claim 7.

12. As noted earlier, run-length encoding is well known in the art. For the sake of illustration, run-length encoding and its applicability to claims 7-11are briefly described here. The examiner takes Official Notice that the following provides a generic description of the run-length encoding method, as it is generally known in the art. Run-length encoding involves reading a sequence of symbols or characters (characters and symbols will be used interchangeably henceforth in this document) which constitutes the data to be compressed and generating an output sequence of symbols, where repeating symbols are replaced by a set of symbols indicating the symbol being repeated and the number of times that symbols is repeated. When a sequence of repeating symbols, called a *run*, is encountered, it is typically encoded into two symbols (which may or may not come from the set of symbols from which the input symbols are an element of). The first symbol typically indicates the length of the sequence of repeating symbols, referred to as a *run-count*. The second symbol typically indicates the symbol that is repeated, referred to as a *run-value*. It is also common to supplement these two symbols with a means to indicate whether the following symbols represent a run (i.e. the following symbols are a run-count and run-value) or the following symbols are un-encoded literal data. More specifically, run-length encoding generally involves reading each

character from the input sequence of characters sequentially and determining, per character, whether or not the character currently being read matches the character immediately preceding it. If the preceding character (call this the read first character[•]) matches the character currently being read (i.e. the character immediately subsequent to the read first character[•]) then the run count (or, if necessary, a corresponding numerical representation thereof) is incremented by one and the process is repeated on the next character. If there is no such match, then the character currently being read is considered a literal, the run-count is reset to zero and the process repeats on the next character without incrementing the run count. Furthermore, if a run is determined to have ended (i.e. when the preceding character does not match the character currently being read and the run count has a value greater than zero), that run is encoded in the manner described above. Literal characters, on the other hand, are written to the output sequence of characters in the form in which they were received, or replaced in that sequence with a corresponding predefined code symbol.

13. It would thus be apparent to one of ordinary skill in the art that run-length encoding addresses most aspects of claim 7, particularly when viewed in light of the applicant's specification. The examiner takes Official Notice that the applicability of run-length encoding as a method for compressing image information is well known in the art. Naturally, as a method of image compression, it would receive as input a sequence of characters or symbols (e.g. bytes) that represent an image. In the manner described above, run-length encoding further involves reading a first character in that sequence[•], reading one or more characters occurring immediately subsequent to the first character in the received sequence of characters, and determining whether or not there is a sequence of matching characters that follow immediately subsequent to the read first character. If there is such as sequence, another sequence of characters is generated which represents the sequence of one or more matching characters. Indeed, the method of compressing image data that is put forth in claim 7 can be reasonably construed by one of ordinary skill in the art as essentially being a run-length encoding scheme. Unlike, the method proposed by claim 7, however, generic implementations of run-length encoding will not include a predefined compression code in the generated sequence representing the matching one of more characters.

[•] Note that denoting this character as the "read first character" is valid because relative to the sequence formed by it and the characters that subsequently follow (regardless of whether there are any) it is, indeed, the first character read. Thus, the distinction of read first character may be given, in this manner, to any arbitrary character in the input sequence of characters. The terminology is used in this manner merely as a means to convey to the reader the inherent similarity between run-length encoding and the compression scheme proposed in claim 7.

14. It is thus evident, that the run-length encoding method taught by Notenboom – as a variant of the generic run-length encoding method discussed above – addresses, in the same manner, the aspects of claim 7addressed in the preceding paragraph. Moreover, in addition to employing run-length encoding in his data compression method, Notenboom suggests the usage of unique flags to replace characters which repeat frequently in the input sequence of characters. (See Notenboom, column 3, lines 33-46). The said flag, used in this manner, can be interpreted as being equivalent, in principle, to the predefined compression code of the applicant's disclosure. Furthermore, Notenboom also provides a motivation for using predefined compression codes or the said flag. According to Notenboom (Notenboom, column 3, lines 44-46), using these codes or flags would avoid the need to provide the character and would thus save one byte in the compressed text (for each encountered run of characters having been assigned a unique flag). It would be clear to one of ordinary skill in the art that this would, in turn, improve the overall compression ratio that can be achieved by the compression method.

15. Claim 7 and the relevant portions of the applicant's specification further propose that the first character of the input sequence of characters is discriminated as representing one of a black portion of the image or white portion of the image. In providing a unique flag(s) which corresponds to a frequently repeating character(s), the compression method of Notenboom implicitly entails the discrimination between characters that have corresponding unique flag and those that do not. Since these unique flags are reserved for frequently repeating characters (Notenboom, column 3, lines 42-43), the compression method of Notenboom would, in turn, entail the discrimination between characters that repeat frequently and those that do not. It can be interpreted from the applicant's disclosure (e.g. page 10, line 18-22 of applicant's disclosure) that much of the image data that will be input into the applicant's claimed compression method will be either solid black or solid white. In addition, the applicant's claimed compression method provides for compression of both black and white image data as well as color image data (see, for example, page 6, lines 16-21of applicant's disclosure). Thus, given the apparent frequency with which black and white image data is expected to occur in the input image data, it would be obvious to one of ordinary skill in the art to provide unique flags for the characters representing black image data and for characters representing white image data, in the manner suggested by Notenboom. In this regard, Notenboom's compression method, if applied to image data, could discriminate between characters representing black or white image data and

characters that do not. Again, this discrimination can be achieved simply by the proper assignment of unique flags to characters representing black image data and characters representing white image data.

16. Although these teachings of Notenboom relate to textual data, they can be trivially extended to treat image data, especially given the well known applicability of run-length encoding to image data compression (see paragraph 13 above). Therefore, taking into account the following:

i. inherent similarity between run-length encoding and the method of compression set forth in claim 7 (as illustrated above) – especially with respect to how the claimed method is further described in the applicant's specification,

ii. Notenboom's usage of predefined compression codes within a run-length encoding compression method, in a manner analogous to what is proposed in claim 7,

iii. the implicit capability of Notenboom's compression method to distinguish between characters having a predefined compression code and those that do not,

iv. and the well known applicability of run-length encoding as a method of image data compression,

it would have been obvious to one of ordinary skill in the art, at the time of the applicant's claimed invention, to apply Notenboom's compression method as a means for compressing image data in the manner claimed in claim 7.

17. Claim 8 merely introduces an additional predefined compression code into the method of claim 7 so as to provide a predefined compression code for characters that represent white image data and another predefined compression code for characters that represent black image data. Thus, according to claim 8, as it is interpreted here, an input sequence is received and a first character (the footnote on page 10) is read. It is determined whether or not that character represents black image data or white image data or not black and not white image data. One or more characters immediately subsequent to the read first character are read. If a sequence of those characters immediately subsequent to the first read character matches the first read character, a new sequence of characters is generated to represent the matching sequence of characters. This newly generated sequence is such that, if the read first character represents white image data, the newly generated sequence will contain the first predefined compression code corresponding to white image data, and, if the read first character represents black image data, the newly generated sequence of characters will contain the second predefined compression code corresponding to black image data. As described above (see paragraph 15 of this document), Notenboom's compression method provides predefined

compression codes for frequently repeating characters. If applied to image data compression, in the manner described above, predefined compression codes could be provided for characters representing black image data and characters representing white image data. It would be apparent to one of ordinary skill in the art that the resulting encoded sequence of characters generated to represent the aforementioned sequence of matching one or more characters would be a sequence of characters containing a first predefined compression code representing a character corresponding to white image data, if the read first character corresponded to white image data; and, the resulting encoded sequence of characters generated to represent the aforementioned sequence of matching one or more characters would be a sequence of characters containing a second predefined compression code representing a character corresponding to black image data, if the read first character corresponded to black image data. Thus, in the manner described previously in paragraphs 11-16 of this document, it would have been obvious to one of ordinary skill in the art, at the time of the applicant's claimed invention, to apply Notenboom's compression method as a means for compressing image data in the manner claimed in claim 8.

18. As mentioned above, step 14 of Notenboom's compression method (Notenboom, Fig. 1, reference number 14) involves run-length encoding all runs consisting of between 4 and 255 identical characters. See Notenboom, column 3, lines 34-36. In this regard, the number 4 represents a predefined lower threshold, whereby the sequence of matching characters is represented by an encoded sequence of characters as per claim 7, only if the number of characters in the sequence of matching characters is greater than or equal to the said lower threshold. Providing such a lower threshold and imposing the constraint that the sequence of matching characters is represented by an encoded sequence of characters as per claim 7, only if the number of characters in the sequence of matching characters is greater than or equal to the said lower threshold, implicitly entails determining if the number of matching characters (i.e. the run count) is greater than or equal to that lower threshold. Thus, Notenboom suggests the usage of a threshold, in an identical manner described in claim 9 and claim 10 of the applicant, within a run-length encoding scheme shown to be inherently similar to the compression method of claim 7 of the applicant. Given this and the arguments made above regarding claim 7, it would have been obvious to one of ordinary skill in the art, at the time of invention, to apply Notenboom's compression method as a means for compressing image data in the manner claimed in claim 9 and 10.

19. Regarding claim 11, it should be clear from the description of run-length encoding above that the sequence of characters that is generated to represent the sequence of one or more matching characters that follow immediately from the read first character will contain some symbol representing the run-count. Thus, Notenboom's compression method, in performing run-length encoding on the input sequence of characters, will generate a sequence of characters to represent the sequence of one or more matching characters that follow immediately from the read first character, which contains a value corresponding to the number of characters in the sequence of one or more matching characters. Given this and the arguments made above regarding claim 7, it would have been obvious to one of ordinary skill in the art, at the time of invention, to apply Notenboom's compression method as a means for compressing image data in the manner claimed in claim 11.

20. Note that the encoder put forth in claim 1 embodies all aspects of the method of claim 7. Therefore, with respect to claim 1, arguments analogous to those presented for claim 7, are applicable (see paragraphs 11-16 above).

21. Note that the encoder put forth in claim 2 embodies all aspects of the method of claim 8. Therefore, with respect to claim 2, arguments analogous to those presented for claim 8, are applicable (see paragraph 17 above).

22. Note that the encoder put forth in claim 3 embodies all aspects of the method of claims 9 and 10. Therefore, with respect to claim 3, arguments analogous to those presented for claim 9 and 10, are applicable (see paragraph 18 above).

23. Note that the encoder put forth in claim 4 embodies all aspects of the method of claim 11. Therefore, with respect to claim 4, arguments analogous to those presented for claim 11, are applicable (see paragraph 19 above).

24. Note that all aspects of the encoder put forth in claim 19 are addressed by the method of claim 7. Therefore, with respect to claim 19, arguments analogous to those presented for claim 1, are applicable (see paragraphs 11-16 above).

25. Note that all aspects of the encoder put forth in claim 20 are addressed by the method of claim 8.Therefore, with respect to claim 20, arguments analogous to those presented for claim 8, are applicable (see paragraph 17 above).

26. Claims 5, 6, 12, and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Notenboom as applied to claim 7 above, and further in view of Burrows (U.S. Patent 5724033) and Lee, et al. (U.S. Patent 5617385). As shown above in paragraphs 11-16 of this document, Notenboom's teachings, when viewed in light of

that which is well known in the art, addresses all aspects of the method of compression claimed in the applicant's claim 7. Notenboom further shows the usage of fixed length run-length encoded sequences to represent the sequences of repetitious characters contained in the input sequence of characters. As stated previously, in the one embodiment of Notenboom's method of compression a generic run-length flag is provided for runs of any character, and the next byte is the repetition count and the next byte is the repeating character (Notenboom, column 3, lines 39-42). Thus, the repetition count and the repeating character, which together represent two of the three elements comprising the encoded sequence that represents the sequence of matching characters, have a total fixed length of two bytes. Although not explicitly stated, the aforementioned generic run-length flag is also one byte. See Notenboom, column 5, lines 17-43 and Fig. 1, reference number 20. There, Notenboom describes a process where bytes of a twice-compressed text (where the first compression applied is run-length encoding compression described in detail above) are replaced with a unique bit string using a technique called Huffman encoding (Notenboom, column 4, lines 59-61 and Fig. 1, reference number 20). In Notenboom, column 5, lines 17-43, Notenboom suggests the application of this technique on the aforementioned run-length flag. Thus, by implication, the generic run-length flag has the length of one-byte. As a result, the sequence of symbols used to represent the sequence of matching characters has a fixed length of three bytes and, hence, a fixed bit-length of 3 × 8 = 24 bits.

27. Notenboom, however, does not show or suggest the usage of a continuation bit, nor the practice of further representing the matching one or more characters by an additional sequence of characters, in the manner of claims 12 and 13 and the relevant portions of the specification. First, it should be noted that the usage of a continuation bit, flag or code, in the manner suggested by claims 12 and 13 and the applicant's specification – that is, to indicate that data contained in a byte (or other data type) continues into other bytes (or other data type), where these bytes (or other data type) are "strung" together to represent data that may otherwise not be representable by a single byte (or other data type), due to the size limitations attributed to that data type – is well known. For example, Burrows shows a method for encoding so-called delta values, which have variable lengths ranging from one byte to multiple bytes. For each delta value which can be encoded as a single byte, a logical zero is stored in the least significant bit of the single byte, and the delta value is stored in the most significant bits of the single byte. Otherwise, for each delta value which must be encoded as a plurality of bytes, a logical one is stored in the least significant bit of the first byte of the plurality of bytes, and a first portion of the delta value is stored in the most significant bits of the first byte. In

this case, a logical zero is stored in the most significant bit of the next byte, and a next portion of the delta value is stored in the least significant bits of the next byte. If the next portion is the last portion of the delta value, a logical zero is stored in the most significant bit of the last byte, and the last portion of the delta value is stored in the least significant bits of the last byte. See Burrows, Abstract, Fig. 8, and column 10, lines 47-59. It would be clear to one of ordinary skill in the art that the most significant bit (MSB) of the first byte of a delta value that spans multiple bytes represents a continuation code, essentially identical to that of claims 12 and 13 and the applicant's specification. It would also be clear that the least significant bit (LSB) of the subsequent bytes of a delta value that spans multiple bytes also represents a continuation code. Furthermore, the bytes that comprise a delta value, which spans multiple bytes, supplement each other in the representation of the delta value. This is very similar to the discussion in the applicant's specification, pertaining to claims 12 and 13, regarding the spanning of large "repeat count" (synonymous with run-count) across multiple 9, 10, 11, or 12 bit blocks (see page 17, lines 28-35 and page 18, lines 1-29 of the applicant's disclosure).

28. Since the delta value of Burrows and the run-count used in run-length encoding both numerically represent integral data, it would be straightforward for one of ordinary skill in the art to use the teachings of Burrows presented above in a run-length encoding method, such as that of Notenboom, to allow for a run-count that spans multiple bytes. It should be noted here that, while Burrows' teachings are limited to data which spans multiple bytes, it would be straightforward for one of ordinary skill in the art to extend these teachings to other data types, such as the 9-12 bit blocks described by the applicant, so that one would obtain a way of representing large values that can span multiple 9, 10, 11, or 12 bit blocks. Also, the placement of the continuation code within each of these bytes or blocks is arbitrary and would be predicated on design considerations (e.g. the endian-ness of the system on which the compression method runs). By incorporating the teachings of Burrows into those of Notenboom and that which is well known in the art, in the manner just described, one obtains a method of compression as per claim 7, where the sequence of characters generated to represent the sequence of one or more matching characters contains a predefined compression code representing frequently repeating characters (e.g. characters that represent black image data and characters that represent white image data) and a run-count that may or may not span a plurality of fixed-length multi-bit blocks, where each of these blocks each contain a continuation code indicating that additional data resides in subsequent bytes or blocks. This is consistent with claims 12 and 13 in the following way. The sequence

of characters generated to represent the sequence of one or more matching characters contains a predefined compression code, representing frequently repeating characters, and one or more fixed-length multi-bit blocks that comprise the run-count. One could view the predefined compression code and the first of these blocks as the second sequence described in claim 7, 12 and 13, which represents (partially) the aforementioned sequence of one or more matching characters, and further includes a continuation code. The subsequent blocks that comprise the run-count would, in turn, constitute the third sequence of characters described in claims 12 and 13, which exclude the predefined compression code, and further represent the aforementioned sequence of one or more matching characters. As shown in Burrows, the aforementioned one or more blocks together represent the value which spans them – namely, the run-count. Thus, the second sequence and third sequence, constructed in the manner just described, can be combined (by combining the said one or more blocks, in the manner shown in Burrow) to obtain a sequence that completely represents the aforementioned sequence of one or more matching characters.

29. While Burrows discloses a method amenable to representing large run-counts, he does not suggest or supply a motivation for applying this method to the run-count found in run-length encoding methods, such as Notenboom's or the applicant's. Lee et al., on the other hand, teach the application of an essentially identical method on run-length data within the context of compressed image storage. See Lee, et al. column 6, lines 6-13 and Fig. 7. In Lee et al., image data is compressed into two-byte pixel data (Lee, et al. Fig. 7, reference number 48) and a series of two-byte run-length data (Lee, et al. Fig. 7, reference number 50). Each two-byte pixel data comprises a one-bit start bit and a 15-bit RGB555 code, and each two-byte run-length data comprises a one-bit continuation bit and a 15-bit run-length code. Note that run-length data is taken to be identical in meaning to the data contained in the run-count. Also note, that, since the usage of a continuation bit, in the manner of Burrows and the applicant, is well known, it can be construed as serving the same purpose in Lee, et al., though they do not expressly define its purpose. The run-length data clearly spans multiple bytes, in a manner similar to that of Burrows and that which is described above. Thus, Lee, et al. would clearly show to one of ordinary skill in the art, at the time of the applicant's claimed invention, the application of a continuation bit and a method inherently similar to that of Burrows (as described above) to the representation of the run-length data, or run-count, present in run-length encoding schemes, such as the one of Notenboom. Therefore, given the teachings and motivations provided by Nootenboom, Burrows, and Lee, et al. and discussed in detail in paragraphs 26-28 of this document, it would have been obvious to one or

ordinary skill in the art, at the time of the applicant's claimed invention, to combine those teachings, in the manner described in paragraphs 26-28 above, to obtain a method of image data compression similar to that which is claimed by the applicant in claims 12 and 13.

30. Note that the encoder put forth in claim 5 embodies all aspects of the method of claim 12. Therefore, with respect to claim 5, arguments analogous to those presented for claim 12, are applicable (see paragraphs 26-29 above).

31. Note that the encoder put forth in claim 6 is embodies all aspects of the method of claim 13. Therefore, with respect to claim 6, arguments analogous to those presented for claim 12, are applicable (see paragraphs 26-29 above).

32. Claims 14-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anzai (U.S. Patent 6009242), in view of Notenboom, in further view of Campbell, et al. (U.S. Patent 5479587).

33. The imaging systems of claims 14 and 15 represent typical imaging systems, where an image processing system (e.g. a RIP) generates compressed image data and transfers that data to a imager controller, where it is, in turn, decompressed and displayed by an imager in a manner peculiar to that imager or type of imager. Such imaging systems abound in prior art. The imaging systems of claims 14 and 15 are distinguished by their modes of compression and decompression. However, as will be illustrated below, it would have been obvious to one of ordinary skill in the art to incorporate such modes of compression and decompression into an imaging system similar to those just described. It was shown above that the compression method and the RIP implementing this method could have been straightforwardly contrived and applied to the compression of image data by one of ordinary skill in the art, given the teachings of Notenboom and that which was well known in the art. It would also be clear to one of ordinary skill in the art that having an RIP that generates compressed image data necessitates a means or method in which to decompress that compressed image data. Thus, by having an imaging system comprised of an RIP that implements the run-length encoding compression method taught by Notenboom (see paragraphs 11-16 above), one is motivated by practical necessity to provide that imaging system with a corresponding decompression method.

34. Anzai discloses an apparatus for outputting image data to a printer comprising a print controller (Anzai, Fig. 1, reference number 7) and a print mechanism (Anzai Fig. 1, reference number 8). The print controller consists

of an input section, a ROM, a CPU, a RAM and an output section (Anzai, Fig. 1, reference number 2, 3, 4, 5 and 6, respectively). An external apparatus such as a host computer supplies input data (e.g. page description language consisting of character codes and control codes). The input section receives this data and outputs it to the CPU. The CPU executes various control programs stored in the ROM. The RAM is assigned as, for example, a frame memory for storing output data (bitmap image data) processed by the CPU. The output section transmits bitmap image data (compressed or not compressed) to the print mechanism. See Anzai, column 5, lines 25-67 to column 6 lines 1-17. Thus, the print controller of Anzai represents what is generally considered in the art as a raster image processor. That is, a device which receives, as input, a stream of printing instructions (e.g. a page description language) and converts it into a bitmap, or raster, image suitable for printing. Furthermore, the CPU constitutes a means for compressing output raster image data on the basis of a data compression program stored on the ROM (Anzai, column 7, lines 5-9).

35. The print mechanism comprises a printer engine for performing a print operation on, for example, a paper sheet, and its controller. The print mechanism executes paper feed processing, print processing, convey processing, and paper delivery processing in accordance with a transmission instruction of output data from the output section. The print mechanism includes a data expansion section (not shown; which may be realized in either a hardware or software manner). When it is determined that output data transferred from the print controller via the interface I/F is compressed data, the print mechanism expands the compressed data to predetermined output data to perform the print processing. See Anzai column 5, lines 57-67 to column 6, lines 1-12. The print mechanism is, therefore, functionally similar to the image controller described above and in claims 14-15 of the applicant. Thus, the print apparatus of Anzai constitutes an imaging system consisting of an RIP capable of generating compressed raster image data and transferring that data to an image controller where it is decompressed and processed for printing. Anzai, however, does not discuss the usage any particular compression method in this print apparatus. As a result, Anzai does not address the aspects of the applicant's claims 14-15 regarding the compression and decompression of image data.

36. Campbell, et al. disclose an imaging system of similar design and functionality to that of Anzai, which employs a run-length encoding method called mode-m compression as a means for image compression. See Campbell, et al. Fig. 1 and the section called "PRINTER SYSTEM" (columns 5-6). Thus, Campbell, et al. suggest

the usage of a run-length encoding method in an imaging system similar to Anzai and applicant's claims 14-15. Given the inherent architectural similarities between the imaging systems of Anzai and Campbell, et al., it would be straightforward for one of ordinary skill in the art to utilize a run-length encoding method as a means for compressing image data (and, consequently, a corresponding run-length decompression method to decompress compressed image data) in the print apparatus of Anzai. Advantages of using such compression/decompression methods include, for example, the fact that they are lossless, extremely fast, and can be employed on-the-fly by a laser printer (Campbell, et al. column 9, lines 48-53). Thus, provided the clear advantages of using run-length compression/decompression methods over other compression/decompression methods and the simplicity with which run-length compression/decompression methods can be incorporated into systems such as Anzai's, it would have been obvious to one of ordinary skill in the art, at the time of the applicant's claimed invention, to utilize a run-length encoding method as a means for compressing image data (and, consequently, a corresponding run-length decompression method to decompress compressed image data) in the print apparatus of Anzai. Note that by combining the teachings of Campbell, et al. with those of Anzai, one could construct an imaging system consisting of an RIP and imager controller, where the RIP produces run-length compressed image data and transfers it to the coupled imager controller (e.g. Anzai's print mechanism), where it is decompressed using a corresponding run-length decompression procedure. While Campbell, et al. do suggest and demonstrate the usage of run-length compression/decompression in an imaging system similar to Anzai and the applicant, they do not teach the method of compression/decompression described in the applicant's claims 14-15.

37. The method of compression utilized by the RIP of claims 14 and 15 is addressed by Notenboom and has been discussed in detail above. Furthermore, the raster image processor (RIP), as claimed in the applicant's claim 14 and claim 15 is functionally similar to the encoder claimed in the applicant's claim 1. Therefore, with respect to the raster image processor of claims 14-15, arguments analogous to those presented for claim 1, are applicable (see paragraph 20 above). The decompression method utilized by the imager controller of claims 14 and 15 are treated below.

38. As with run-length encoding methods, run-length decompression methods are well known in the art. These are essentially the inverse of the run-length encoding compression method described above and will not be treated here in any detail. As mentioned above, typical run-length encoding methods do not make use of predefined

compression codes in the manner discussed in applicant's disclosure. Similarly, typical run-length decompression methods do not employ these codes. However, Notenboom shows the run-length decompression of run-length encoded sequences that do include such predefined compression codes. Note that these run-length encoded sequences codes would have been generated in the manner illustrated in detail above and in Notenboom. See Notenboom, column 8, lines 3-21 and Fig.2b, reference numbers 44, 46, 54, 56, 52, 58, and 60. With regard to Fig. 2b, it would have been obvious to one of ordinary skill in the art that steps 42, 48, and 50 have nothing to do with run-length encoding or run-length decompression. Furthermore, one of ordinary skill in the art would arrive at the same conclusion for the portions of steps 46, 54 and 56 dealing with "Key Phrase Flag" and decompressing "Bit Codes"[†]. Again, it should be emphasized that the essential teachings of Notenboom with regard to the applicant's claims are those that pertain to run-length encoding and the corresponding decompression method. Input to step 44, is a symbol (a text character or flag) from the encoded sequence of characters that represent the original input sequence of characters. It is determined in step 44 whether that symbol is a text character or a flag. If the symbol is a text character then it is written to the decompressed sequence of characters in step 52. This is consistent with typical run-length decompression methods. If, in step 44, the symbol is a flag it is then determined whether that flag is a Key Phrase Flag or a Run Length Flag in step 46. Again, as previously noted, one of ordinary skill in the art seeking to construct a run-length decompression method would not require this determination since key phrase flags, as they are defined by Notenboom, would not be considered in such a method. Next, in step 54, if the run-length flag is a run flag of an arbitrary character the next symbol is read and the run count is determined from it and saved. The next symbol is read and the repeated character is determined from it. Then, in step 52, that character is then written to the decompressed sequence of characters repeatedly for the number of times indicated by the saved run count. This

[†] One of ordinary skill in the art, seeking to construct a decompression method that was essentially the inverse of the run-length compression method that was described in the preceding paragraphs, would ignore the references to "key phrase flags" and "decompressing bit codes", since they do not pertain to run-length encoding. These relate to subsequent disjoint stages of compression illustrated in Fig.1 in Notenboom. Any of these stages can be removed without affecting the functionality of the remaining compression stage(s). Nootenboom implies the functional independence of these compression stages in column 2, line 65-68 to column 3, lines 1-6. The helpmake program which embodies the steps shown in Fig. 1 of Notenboom allows the user to select the amount of compression to take place. In particular, a user may indicate run-length compression only (Notenboom column 2, lines 66-67). This implies that the functionality of the run-length compression scheme taught by Notenboom is not connected to the operation of the compression that takes place in the subsequent stages shown in Fig. 1. One of ordinary skill in the art could further conclude, due to the relationship of the run-length compression method and the run-length decompression method, that the run-length decompression can be viewed as functionally independent from the other methods of decompression shown in Fig. 2b.
Thus, in the subsequent portions of this document it is assumed that "key phrase flags" and "bit codes" are not used, since they are not pertinent to run-length compression or decompression. Furthermore, since bit codes and key phrase flags are not used, it would be obvious to one of ordinary skill in the art that the data presented to step 44 is a run-length encoded sequence of symbols. Since these symbols are available without decompressing bit codes – that is, the symbols can be read literally from the run-length encoded sequence of symbols – steps 42, 54, and 56 will involve reading a symbol where it is indicated in Fig. 2b to decompress bit codes.

again is consistent with typical implementations of run-length decompression. If the run-length flag, on the other hand, is one of the unique flags given to characters that repeat frequently (see Notenboom column 3, lines 42-46), the next symbol is read and the run count is determined from it. Then, in step 52, the corresponding character which that unique flag represents is written to the decompressed sequence of characters repeatedly for the number of times indicated by the saved run count. If there are additional symbols to be read from the input encoded sequence of characters, the process repeats. If not the decompression process ends.

39. Although these teachings of Notenboom relate to run-length encoded textual data, they can be straightforwardly applied to run-length encoded image data, given the well known applicability of run-length encoding as a method of image data compression. In the manner suggested in paragraphs 14-15 of this document, the unique flags of this decompression procedure can be interpreted as being equivalent, in principle, to the predefined compression codes of the applicant's disclosure. If this decompression method were applied to sequences of symbols representing compressed image data, it would have been obvious to one of ordinary skill in the art to assign a unique flag to symbols corresponding to white image data and a unique flag to symbols corresponding to black image data, especially if the image data were expected to contain large sections of black and/or white image data. Clearly, in this regard, the run-length decompression taught by Notenboom, when viewed in light of that which is well known in the art, additionally addresses all aspects of the decompression method described in claims 14 and 15.

40. Since the mode-m compression/decompression methods described by Campbell, et al. and the run-length compression/decompression methods or Notenboom are both variants of run-length compression/decompression, employing the latter in the imaging system obtained by combining the teachings of Anzai and Campbell, et al. would have been straightforward to one of ordinary skill in the art. In doing so, one would obtain an imaging system with the advantages discussed in paragraph 14 of this document. Taking this into account and the simplicity with which the modification can be performed, it would have been obvious to one ordinary skill in the art to employ the run-length compression/decompression method of Notenboom in the imaging system obtained by combining the teachings of Anzai and Campbell, et al. By combining the teachings of Anzai, Campbell, et al., and Notenboom, in the manner described above, one would obtain an imaging system similar to that of the applicant's claims 14 and 15.

41. The raster image processor (RIP), as claimed in the applicant's claim 16 is functionally similar to the encoder claimed in the applicant's claim 2. Therefore, with respect to the raster image processor of claims 16, arguments analogous to those presented for claim 2, are applicable (see paragraph 21 above). Furthermore, since the run-length decompression method of Notenboom (see paragraphs 38-39 above) can, in principle, interpret multiple predefined compression codes, one of ordinary skill in the art can, in same manner described above in relation to claims 14 and 15, obtain an imaging system similar to that of claim 16. Therefore, arguments analogous to those presented for claims 14 and 15 are applicable to claim 16 (see paragraphs 33-40).

42. The raster image processor (RIP), as claimed in the applicant's claim 17 is functionally similar to the encoder claimed in the applicant's claim 3. Therefore, with respect to the raster image processor of claims 16, arguments analogous to those presented for claim 3, are applicable (see paragraph 22 above). For all other aspects of claim 17 arguments analogous to those presented for claims 14 and 15 are applicable (see paragraphs 33-40).

43. The raster image processor (RIP), as claimed in the applicant's claim 18 is functionally similar to the encoder claimed in the applicant's claim 4. Therefore, with respect to the raster image processor of claims 16, arguments analogous to those presented for claim 4, are applicable (see paragraph 23 above). For all other aspects of claim 18 arguments analogous to those presented for claims 14 and 15 are applicable (see paragraphs 33-40).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kevin Siangchin whose telephone number is (703)308-6604. The examiner can normally be reached on 9:00am - 5:30pm, Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Amellia Au can be reached on (703)308-6604. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)306-0377.

Kevin Siangchin
Examiner
Art Unit 2623

AMELIA M. AU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600